Utility Patent Application
CTX-057
(1545/104)

A SYSTEM AND METHOD OF EXPLOITING THE SECURITY OF A SECURE COMMUNICATION CHANNEL TO SECURE A NON-SECURE COMMUNICATION CHANNEL

FIELD OF THE INVENTION

The invention relates generally to client-server computer networks. More specifically, the invention relates to a system and method for securely accessing software applications using a remote display protocol.

BACKGROUND OF THE INVENTION

Software applications that are requested to be remotely displayed on a client computer, or client, are commonly accessed with a graphical or windowing terminal session. When a user requests an application on a client computer, the application executes on a server and typically the input information (e.g., mouse and keyboard information) and display information are transmitted from the server computer to the client computer. Graphical or windowing terminal sessions often make use of unauthenticated connections between the client and the server. Alternatively, the graphical or windowing terminal session may authenticate the connection between the client and the server with the user supplying his password to the server.

The aforementioned techniques employed by the terminal sessions have various shortcomings. For example, transmitting information, such as password information, to an unauthenticated server allows the information to be viewed by a server that is not trusted by the client. The non-secure connection permits an eavesdropper to intercept a user's password for future use.

To avoid these problems, the client and server are typically authenticated using conventional cryptographic techniques. One type of cryptographic technique used by networks is a ticket-based authentication scheme. Most current ticket-based authentication schemes transmit a ticket. The ticket, which can typically be used only one time, may contain an encryption key to be used in future communications and/or may contain a secret password to support the future communications. When the client and the server both have the encryption key, they can communicate securely.

However, the current ticket-based authentication schemes are limited in several areas. First, the ticket is typically transmitted to the client over a non-secure communication channel, thereby allowing an eavesdropper to intercept the ticket and retrieve the encryption key. Using the encryption key, the eavesdropper can pose as the server to the client or as the client to the server. Second, the current schemes do not take advantage of secure web pages. For example, current ticket-based authentication schemes make transactions over the internet, such as purchases, unsafe because proprietary information, such as a purchaser's credit card information, can be transmitted to a non-secure web page. Third, software applications executing on a server are commonly transmitted over a non-secure communication channel for display on a remote display protocol on a client machine. For instance, networks may consist of specialized application servers (e.g., Metaframe for Windows, manufactured by Citrix Systems, Inc. of Ft. Lauderdale, Florida), to execute specific applications which are typically transmitted to a remote display service over a non-secure communication channel. Fourth, although the ticket can typically be used only one time (i.e., making it a "one-time use" ticket) and having no further value after its first use, the one-time use ticket does not protect the user's password (which is used for login into an operating system or an application) from an eavesdropper on the ticket's

first transmission. Therefore, the user's password is still not completely protected from interception and the server is consequently not authenticated to the client.

SUMMARY OF THE INVENTION

The present invention features a system and method for establishing a secure communication channel between a client and an application server. A ticket service generates a ticket having an identifier and a session key. A communications device obtains the ticket from the ticket service and transmits the ticket to a client over a secure communication channel. The client transmits the identifier of the ticket to an application server over an application communication channel. The application server then obtains a copy of the session key of the ticket from the ticket service. Communications exchanged between the client and the application server over the application communication channel are then encrypted using the session key to establish the application communication channel as a secure communication channel.

In one embodiment, a web browser executing on a client establishes communications with a web server over a secure web communication channel. The client receives a ticket having an identifier and a session key from the web server over the secure web communication channel. The client then transmits the identifier of the ticket to the application server over the application communication channel to provide the application server with information for obtaining a copy of the session key.

In one aspect, the invention relates to a method for establishing a secure communication channel between a client and an application server. The client receives a ticket having an identifier and a session key from a web server over a secure web communication channel. The client then transmits the identifier of the ticket to the application server over an application communication channel to provide the application server with information for obtaining a copy

of the session key. The client establishes a secure communication channel over the application communication channel by using the session key to encrypt and decrypt communications to and from the application server. The identifier is a nonce. In one embodiment, the client and the web server use secure socket layer technology to establish the secure web communication channel.

In another aspect, the invention relates to a communications system that establishes a secure communication channel. The communications system includes a client, an application server, a communications device, and a ticket service. The ticket service generates a ticket having an identifier and a session key. The communications device is in communication with the ticket service to obtain the ticket. The client is in communication with the communications device over a secure communication channel to receive the ticket from the communications device. The application server is in communication with the client over an application communication channel to receive the identifier of the ticket from the client and in communication with the ticket service to obtain a copy of the session key from the ticket service. The application server and the client exchange communications over the application communication channel as a secure communication channel. In one embodiment, the ticket service resides on the communications device. In one embodiment, the communications device is a web server.

DESCRIPTION OF THE DRAWINGS

The aspects of the invention presented above and many of the accompanying advantages of the present invention will become better understood by referring to the included drawings, which show a system according to the preferred embodiment of the invention and in which:

Fig. 1 is a block diagram of an embodiment of a communication system for establishing secure communications between a client and an application server in accordance with the principles of the invention; and

Fig. 2 is a flow diagram of an embodiment of the communications performed by the communications system shown in Fig. 1 to establish secure communications between the client and the application server.

DETAILED DESCRIPTION

Fig. 1 shows a block diagram of an embodiment of a communication system 100 including a client 10 in communication with an application server 15 over an application communication channel 25 and in communication with a communications device 20 over a communication channel 30. The communication channel 30 and the application communication channel 25 pass through a network 27. In other embodiments, the communication channel 30 and the application channel 25 pass through other, different networks. For example, the communication channel 30 can pass through a first network (e.g., the World Wide Web) and the application communication channel 30 can pass through a second network (e.g., a direct dial-up modem connection). The communication channel 30 is a secure communication channel in that communications are encrypted. The application server 15 is additionally in communication with the communications device 20 over a server communication channel 35. The application server 15 and the communications device 20 are part of a server network 33. By exploiting the security of the secure communications between the client 10 and the communications device 20 over the secure communication channel 30, the communication system 100 establishes a secure communication link over the non-secure application communication channel 25 to remotely display desktop applications securely on the client 10.

The network 27 and the server network 33 can be a local-area network (LAN) or a wide area network (WAN), or a network of networks such as the Internet or the World Wide Web (i.e., web). The communication channel 30 can be any secure communication channel. In one embodiment, the communication channel 30 (hereafter web communication channel 30) supports communications over the web. In one embodiment, the server network 33 is a protected network that is inaccessible by the public. The server communication channel 35 traverses the server network 33 and therefore can be a non-secure communication channel. Example embodiments of the communication channels 25, 30, 35 include standard telephone lines, LAN or WAN links (e.g., T1, T3, 56kb, X.25), broadband connections (ISDN, Frame Relay, ATM), and wireless connections. The connections over the communication channels 25, 30, 35 can be established using a variety of communication protocols (e.g., HTTP, TCP/IP, IPX, SPX, NetBIOS, Ethernet, RS232, and direct asynchronous connections).

The client 10 can be any personal computer (e.g., 286, 386, 486, Pentium, Pentium II, Macintosh computer), Windows-based terminal, Network Computer, wireless device (e.g., cellular phone), information appliance, RISC Power PC, X-device, workstation, mini computer, main frame computer, personal digital assistant, or other communications device that is capable of communicating over the secure web communication channel 30. In one embodiment, the client 10 operates according to a server-based computing model. In a server-based computing model, the execution of application programs occurs entirely on the application server 15 and the user interface, keystrokes, and mouse movements are transmitted over the application communication channel 25 to the client 10. The user interface can be text driven (e.g., DOS) or graphically driven (e.g., Windows). Platforms that can be supported by the client 10 include DOS and Windows CE for windows-based terminals.

In one embodiment, the client 10 includes a web browser 40, such as Internet Explorer™ developed by Microsoft Corporation in Redmond, WA, to connect to the web. In a further embodiment, the web browser 40 uses the existing Secure Socket Layer (SSL) support, developed by Netscape in Mountain View, California, to establish the secure web communication channel 30 to communications devices such as the communications device 20. The web browser 40 also has a user interface that may be text driven or graphically driven. The output of an application executing on the application server 15 can be displayed at the client 10 via the user interface of the client 10 or the user interface of the web browser 40. Additionally, the client 10 includes an application client 41 for establishing and exchanging communications with the application server 15 over the application communication channel 25. In one embodiment, the application client 41 is the Independent Computing Architecture (ICA) client, developed by Citrix Systems, Inc. of Fort Lauderdale, Florida, and is hereafter referred to as ICA client 41. Other embodiments of the application client 41 include the Remote Display Protocol (RDP), developed by Microsoft Corporation of Redmond, Washington, X-Windows, developed by Massachusetts Institute of Technology of Cambridge, Massachusetts, a data entry client in a traditional client / server application, and a Java applet.

The application server 15 hosts one or more application programs that can be accessed by the client 10. Applications made available to the client 10 for use are referred to as published applications. Examples of such applications include word processing programs such as MICROSOFT WORD® and spreadsheet programs such as MICROSOFT EXCEL®, both manufactured by Microsoft Corporation of Redmond, Washington, financial reporting programs, customer registration programs, programs providing technical support information, customer database applications, or application set managers. In another embodiment, the application

server 15 is a member of a server farm (not shown). A server farm is a logical group of one or more servers that are administered as a single entity.

In one embodiment, the communications device 20 (hereafter web server 20) is a computer that delivers web pages to the client 10. In other embodiments, the communications device 20 can be any personal computer (e.g., 286, 386, 486, Pentium, Pentium II, Macintosh computer), Windows-based terminal, Network Computer, wireless device (e.g., cellular phone), information appliance, RISC Power PC, X-device, workstation, mini computer, main frame computer, personal digital assistant, or other communications device that is capable of establishing the secure web communication channel 30 with the client 10.

In one embodiment, the web server 20 also includes a ticket service 60. The ticket service 60 controls communication security. The ticket service 60 generates a ticket containing an encryption key. The ticket is transmitted to the client 10 (i.e., the web browser 40) over the secure web communication channel 30. The transmission of the ticket to the client 10 over the secure web communication channel 30 facilitates the establishment of secure communications over the application communication channel 25 between the client 10 and the application server 15 in accordance with the principles of the invention. In another embodiment, the ticket service 60′ resides on another server 20′. The server 20′ (and ticket service 60′) is in communication with the web server 20 and the application server 15 over a server communication channel 35′. In yet another embodiment, the ticket service 60 is a separate component (not shown) of the server network 33. The web browser 40 then sends the ticket to the ICA client 41. A technique often used to transmit application data from applications executing on the application server 15 over a secure connection to the client 10 is to transmit the application data to the client 10 through the web server 20 over the secure connection between the client 10 and the web server

20. This technique is inefficient in that communication between the application server 15 and the client 10 takes an additional "hop"; namely the web server 20. The present invention uses the ticketing mechanism to establish a secure communication link directly between the application server 15 and the client 10, thereby eliminating the intermediate transmission of application data from the application server 15 to the web server 20.

A client user requesting an application or server desktop, for example, to be remotely displayed on the client 10 first establishes a communication link 32 with the web server 20 over the web communication channel 30 and passes login and password information to the web server 20. In one embodiment, the client user uses the web browser 40 to request an application from the web server 20 that is listed on a web page displayed by the web browser 40.

In a further embodiment, the web browser 40 uses SSL to establish the secure web communication channel 30. To use the SSL protocol to establish the secure web communication channel 30, the web browser 40 or an application executing on the client 10 attempts to connect to a secure web page on the web server 20. The web server 20 then asserts the web server's identity to the client 10 by transmitting a secure web server certificate to the client 10. A certification authority (CA) issues the secure web server certificate to the web server 20. Web browsers 40 have a list of trusted CAs (i.e., public key of the CA) embedded within the software of the web browser 40. The client 10 verifies the web server certificate by decrypting the signature of the CA in the web server's certificate with the public key of the CA embedded in the web browser 40 (or application). Therefore, in order to establish a secure communication channel using SSL, the web browser 40 or the application executing on the client 10 has the public key of the CA embedded in the software prior to attempting to connect to the secure web page. Besides using the SSL protocol to establish the secure web communication channel 30, the

web browser 40 can connect to the web server 20 over the web communication channel 30 using other security protocols, such as, but not limited to, Secure Hypertext Transfer Protocol (SHTTP) developed by Terisa Systems of Los Altos, CA, HTTP over SSL (HTTPS), Private Communication Technology (PCT) developed by Microsoft Corporation of Redmond, Washington, Secure Electronic Transfer (SET), developed by Visa International, Incorporated and Mastercard International, Incorporated of Purchase, NY, Secure-MIME (S/MIME) developed by RSA Security of Bedford, Massachusetts, and the like.

Once the communication link 32 is established, the web server 20 generates a ticket for the communication session. The ticket includes a first portion and a second portion. In one embodiment, the first portion, also referred to as a session identifier (ID) or nonce, is a cryptographic random number that can be used within a certain time period determined by the web server 20. The second portion is an encryption key, hereafter referred to as a session key. The web server 20 stores the ticket in local memory and then transmits (arrow 34) a copy of the ticket to the web browser 40 on the client 10.

In one embodiment, the ticket includes additional information, such as the network address of the application server 15. In another embodiment, the web server 20 independently transmits the address of the application server 15 to the client 10. For example, if the client 10 requests an application by name from the web server 20, the web server 20 converts the application name into the network address of the application. Examples of the additional information included in the ticket are, but not limited to, the time that the ticket is valid, the screen size of the application when displayed on the client 10, the bandwidth limits of the web communication channel 30 and/or the application communication channel 25, and billing information. As described more fully below, the web server 20 also associates the user's login

information, such as the user's password, with the ticket stored in local memory for future retrieval by the application server 15.

The ICA client 41 obtains the ticket from the web browser 40 and subsequently transmits (arrow 42) the session ID (i.e., the first potion) of the ticket to the application server 15. The session ID can be transmitted in encrypted or cleartext form. The application server 15 decrypts the session ID, if encrypted, and transmits (arrow 44) a request to the web server 20 for a session key that corresponds to the session ID received from the client 10. The web server 20 verifies the session ID, as described below, and sends (arrow 48) the corresponding session key to the application server 15 over the server communication channel 35.

Both the application server 15 and the client 10 (i.e., the ICA client 41) now possess a copy of the session key without requiring the transmission of the ticket or the session key over the non-secure application communication channel 25. By using the session key to encrypt and decrypt the communications over the previously non-secure application communication channel 25, the client 10 and the application server 25 establish (arrow 50) a secure communication link 50 over the application communication channel 25. Moreover, the user's login information (e.g., password) is not transmitted between the client 10 and the application server 15 over the non-secure application communication channel 25. Therefore, the present invention strengthens (arrow 50) the security of the communication link 50 over the non-secure application communication channel 25 by not exposing sensitive information, such as the user's password, to eavesdroppers intercepting communications over the non-secure application communication channel 25. Additionally, because the application server 15 and the client 10 communicate with the same session key, they share a secret that was transmitted by the ticket service 60. The ticket service 60 indirectly authenticates the application server 15 and the client 10, and the ticket

service 60 is vouching for each. Therefore, the authentication server 15 and the client 10 perform mutual authentication. In one embodiment, the client 10 again transmits the user's password over the web communication channel 30 to the web server 20 to provide compatibility with legacy systems (e.g., an unmodified operating system login sequence on the web server 20 that requires the client 10 to transmit the user's password multiple times).

In more detail, Fig. 2 shows embodiments of a process performed by the communications system 100 to establish a secure communication link 50 over the application communication channel 25 between the client 10 and the application server 15. The web browser 40 lists (step 200) web links to software applications or server desktops on the web page that the user of the client 10 views. The client user, using the web browser 40, requests (step 205) a software application from the web server 20. In one embodiment, the web browser 40 establishes the secure web communication channel 30 using the previously described SSL protocol. In this embodiment, the client 10 (e.g., the web browser 40) authenticates the web server 20 using a public key (e.g., X509) certificate. In a further embodiment, the client 10 is also authenticated to the web server 20 using a public key certificate.

In another embodiment, the web server 20 authenticates the user when the user uses the web browser 40 to request an application from the web server 20. For example, the web server 20 requests the user's login information, which includes the user's login name and password, with a request displayed on the web browser 40. The user provides (step 210) the user's login information to the web browser 40. The web browser 40 subsequently transmits (step 220) the user's login name and password to the web server 20 over the secure web communication channel 30. In another embodiment, the user's login information is any code or method that the web server 20 accepts to identify the user's account on the web server 20.

The web server 20 transmits (step 230) the user's login information to the ticket service 60. The ticket service 60 verifies (step 240) the user's login information and determines whether the user is entitled to access the requested application. Depending on the declared communication security policy for that application, the ticket service 60 either refuses or grants access to the application by the user. If the ticket service 60 denies access, the web browser 40 displays an HTML error or an error web page on the client 10. When the ticket service 60 grants access to the requested application, the ticket service 60 generates (step 245) a ticket for the session and transmits (step 250) the ticket to the web server 20.

As described above, the ticket includes a session ID and a session key. The session ID can be used once within a certain time period and makes the ticket a "one-time use" ticket having no further value after its first use. The web server 20 then stores (step 253) the ticket in local memory. In a further embodiment, the web server 20 associates the login information provided by the user in step 210 and other security information used to authorize the session, such as the requested application name, with the stored ticket for later retrieval by the application server 15. The web server 20 subsequently transmits (step 255) the ticket to the client 10 over the secure web communication channel 30.

The web browser 40 extracts (step 260) the session ID from the ticket and presents (step 265) the session ID to the application server 15. The application server 15 checks the session ID to ensure that the session ID has not been used previously with this client 10. In one embodiment, the application server 15 monitors (e.g., stores in local memory) each ticket (i.e., session ID) that the client 10 transmits to the application server 15. In another embodiment, the ticket service 60 checks the session ID to ensure that the session ID has not been used previously with this client 10. In yet another embodiment, the ticket service monitors each ticket that the

ticket service 60 transmits to the web server 20 to ensure that each session ID is transmitted to the ticket service 60 only once.

The application server 15 then uses the session ID to determine the session key associated with the presented session ID. To accomplish this, the application server 15 transmits the session ID to the ticket service 60 and requests (step 270) the session key from the ticket service 60 of the web server 20 in response to the session ID. The ticket service 60 accesses local memory and uses the session ID as an index to retrieve the ticket information associated with the session ID. The ticket service 60 then returns (step 280) the session key associated with the session ID to the application server 15.

To increase optimization of the communications between the application server 15 and the web server 20, in an alternate embodiment the web server 20 transmits (shown as phantom step 266) to the application server 15 additional information (e.g., the requested application name, the user's login information) that was previously associated with the ticket in step 253. The application server 15 retrieves (phantom step 267) the additional ticket information and authorizes the communication session from this additional information. This additional information, such as the user's password and/or the name of the requested application, was not transmitted to the application server 15 by the client 10 over the non-secure application communication channel 25, thereby protecting the information from potential attackers. In this embodiment, the application server 15 verifies (phantom step 268) the additional information. If the additional information is not valid, the application server 15 refuses (phantom step 269) access to the requested application by the user. If the additional information is valid, the application server 15 grants access to the requested application and, as described above, requests (step 270) the session key from the ticket service 60.

In another embodiment, the ticket service 60 performs additional checks on the session ID. For example, the ticket service 60 performs checks on the session ID for early detection of replay (i.e., checking that the session ID has not been previously transmitted to the ticket service 60) and/or Denial of Service (DoS) attacks (i.e., flooding and eventually disabling a remote server with illegitimate packets of data). In yet another embodiment, the web server 20 transmits the first and second portion of the ticket to the application server 15 before the application server 15 requests it (step 270), thus eliminating the request in step 270. In this embodiment, the application server 15 stores the session key in its local memory and retrieves from its local memory the session key after the client 10 presents (step 265) the session ID to the application server 15.

After the application server 15 obtains (step 280) the session key, the application server 15 uses the session key to encrypt communications to the client 10 and to decrypt communications from the client 10 over the application communication channel 25. Similarly, the client 10 uses the session key that the client 10 obtained from the ticket transmitted over the secure web communication channel 30 to decrypt communications from the application server 15 and to encrypt communications to the application server 15. Because the client 10 and the application server 15 use the session key to encrypt and decrypt communications over the application communication channel 25, the client 10 and the application server 15 establish (step 290) the secure communication link 50 over the previously non-secure application communication channel 25. Moreover, because the client 10 and the application server 15 have the session key without transmitting the ticket over the non-secure application communication channel 25 (and thus potentially revealing the session key to third parties), the client 10 and the

application server 15 strengthen the security of the communication link 50 over the previously non-secure application communication channel 25.

In one embodiment, the application communication channel 25 is made secure using the SSL protocol. In this embodiment, the ticket service 60 substitutes an application server certificate for the session key in the ticket. The client 10 uses the application server certificate to communicate with the application server 15. The application server certificate is downloaded to the client 10 over the web communication channel 30 in response to a request for the ticket. Therefore, because the application server certificate is downloaded to the client 10 over a secure link (i.e., the web communication channel 30), the application server certificate does not need to be signed by a well-known public CA. Although the client 10 did not have the application server's certificate or the CA key in advance, an authenticated secure connection is established over the application communication channel 25 using the application server certificate included in the ticket.

For example, if the client 10 requests another SSL component (e.g., a separate instance or implementation of the requested software application) and the client 10 does not have the CA certificate in its local memory (e.g., database, local disk, RAM, ROM), the client 10 can use the application server certificate transmitted in the ticket to establish an authenticated secure connection over the application communication channel 25. More specifically, the client 10 uses the application server certificate transmitted in the ticket when the client 10 does not have a CA root certificate stored in its local memory that is associated with the requested SSL component (or when the client 10 has an incomplete list of CA certificates that does not include a CA certificate for the requested SSL component) and the client 10 cannot access the CA database of the web browser 40. Furthermore, because a signed CA certificate is needed for the web server

20 but is not needed for an application server 15 (i.e., each application server 15 that is a member of a server farm), the costs (and overhead) of obtaining the required number of signed CA certificates for secure communication is reduced. In another embodiment, the application server 15 stores a private key for decryption of messages that are encrypted with a corresponding public key. The ticket service 60 consequently transmits the corresponding public key of the application server 15 to the client 10 to encrypt communications.

In this embodiment, the session ID still provides additional value, in that it ensures that the client 10 can gain access to the requested application and can gain access one time because ticket service 60 (or web server 20) monitors the ticket (i.e., the session ID). Furthermore, if the application server 15 and the client 10 use different session keys to encrypt and decrypt communications over the application communication channel 25, an eavesdropper cannot modify the session ID transmitted by the client 10 to the application server 15 because the session ID and the cryptographic checksum do not match the checksum expected by the application server 15 (i.e., integrity check). Therefore, the client 10 and the application server 15 determine when different session keys are used (e.g.,"man-in-the-middle" attack) by the application server 15 and the client 10 to encrypt and decrypt communications over the application communication channel 25.

In a further embodiment, the session key is substantially equivalent to a null value (i.e., the ticket contains only a nonce or a nonce and a constant value for the session key). When the session key is substantially equivalent to a null value, the client 10 does not transmit the user's login information (e.g., password) between the client 10 and the application server 15 over the non-secure application communication channel 25. Therefore, because the ticket is only valid for a single use and only grants access to a previously authorized resource (e.g., the ICA client

41), the external password exposure can be avoided and individual session level access control can be achieved, even with a null or fixed session key value.

Additionally, because no information is pre-configured into the web browser 40 or the client 10 in order to remotely display the requested application (i.e., because the client 10 does not need to be populated with a server certificate or a CA certificate), the present method is a "zero-install" solution for secure access to desktop applications over the web. Further, the web browser 40 receives the ticket and the ICA client 41 from the web server 20 over the communication channel 30. In this embodiment, the web server 20 transmits the ticket and a MIME type document, as described above, specifying that the data includes a "document" for the ICA client 41 (as a helper application). The MIME type document invokes the ICA client 41 and the web browser 40 transfers the ticket to the ICA client 41, thus allowing the exploitation of the security of the communication channel 30 to secure the application communication channel 25 without having the ICA client 41 pre-installed on the client 10. Having described certain embodiments of the invention, it will now become apparent to one of skill in the art that other embodiments incorporating the concepts of the invention may be used. Therefore, the invention should not be limited to certain embodiments, but rather should be limited only by the spirit and scope of the following claims.